

It is claimed:

1. A method for protecting a sequence of computer code comprising the steps of:

preparing obscuring instructions;

injecting a large number of obscuring instructions into the sequence of computer code in an automated process to produce an obscured sequence of computer instructions that in total is humanly impossible to read and understand; and

encrypting a static image of the obscured sequence to protect against direct decompilation.

2. The method of claim 1 further comprising the step of executing the obscured instructions one instruction at a time, thereby making run time trace and observation a labor intensive manual process.

3. The method of claim 1 wherein the obscuring instructions are identified by codes further comprising the steps of:

generating a first set of obscuring instructions having a first set of codes associated therewith;

transforming the first set of codes associated with the first set of obscuring instructions into a second set of codes; and

generating a second set of obscuring instructions identified by the second set of codes.

4. The method of claim 3 wherein the codes are numeric codes and a mathematical transformation is performed on the numeric codes of the first set to produce the numeric codes of the second set.

5. The method of claim 1 wherein the obscuring instructions are identified by numeric codes further comprising the steps of:

generating a first set of obscuring instructions having a first set of numeric codes associated therewith;

performing a mathematical transformation on the numeric codes of the first set to produce a second set of numeric codes; and

generating a second set of obscuring instructions identified by the second set of numeric codes.

5  
6. The method of claim 5 further comprising the step of compressing the static image by recording in the static image a record of the transformation used to generate the second set of obscuring instructions.

7. The method of claim 1 further comprising the step of compressing the static image.

8. The method of claim 1 wherein the obscured sequence of computer instructions is organized into a sequence of blocks of computer instructions and the steps of encrypting a static image comprises the steps of:

encrypting a first block to form a first encrypted output;

encrypting a second block and the first encrypted output to form a second encrypted output; and

encrypting a third block and the second encrypted output to form a third encrypted output.

9. The method of claim 8 further comprising the step of compressing each block as it is encrypted.

10. A method for protecting a data file comprising the steps of:

preparing obscuring data;

injecting a large number of obscuring data into the data file in an automated process to produce an obscured sequence of data that in total is humanly impossible to read and understand; and

encrypting a static image of the obscured sequence to protect against direct decompilation.

11. The method of claim 10 wherein the obscuring data are identified by codes further comprising the steps of:

generating a first set of obscuring data having a first set of codes associated therewith;

transforming the first set of codes associated with the first set of obscuring data into a second set of codes; and  
generating a second set of obscuring data identified by the second set of codes.

12. The method of claim 11 wherein the codes are numeric codes and a mathematical transformation is performed on the numeric codes of the first set to produce the numeric codes of the second set.

13. The method of claim 10 wherein the obscuring data are identified by numeric codes further comprising the steps of:

generating a first set of obscuring data having a first set of numeric codes associated therewith;

performing a mathematical transformation on the numeric codes of the first set to produce a second set of numeric codes; and

generating a second set of obscuring data identified by the second set of numeric codes.

14. The method of claim 13 further comprising the step of comprising the static image by recording in the static image a record of the transformation used to generate the second set of obscuring data.

15. The method of claim 10 further comprising the step of compressing the static image.

16. The method of claim 10 wherein the obscured sequence of data is organized into a sequence of blocks of data and the steps of encrypting a static image comprises the steps of:

encrypting a first block to form a first encrypted output;

encrypting a second block and the first encrypted output to form a second encrypted output; and

encrypting a third block and the second encrypted output to form a third encrypted output.

17. The method of claim 16 further comprising the step of compressing each block as it is encrypted.

18. The method of claim 10 further comprising the steps of:  
generating a first set of obscuring data; and  
performing a mathematical transformation on the first set of obscuring data to produce a second set of obscuring data.

19. Apparatus for protecting a sequence of computer code with obscuring instructions, said apparatus comprising:  
means for automatically injecting a large number of obscuring instructions into the sequence of computer code to produce an obscured sequence of computer instructions that in total is humanly impossible to read and understand; and  
means for encrypting a static image of the obscured sequence to protect against direct decompilation.

20. The apparatus of claim 19 further comprising means for executing the obscured instructions one instruction at a time, thereby making run time trace and observation a labor intensive manual process.

21. Apparatus for protecting a sequence of computer code with obscuring instructions comprising:  
an obscuring instruction bank, for storing obscuring instructions each of which is identified by a code;  
a transformation function bank for storing transformation functions;  
a generator for generating blocks of obscuring instructions by selecting obscuring instructions from the obscuring instruction bank and transformation functions from the transformation function bank, said transformation functions being used by the generator to transform the codes that identify selected obscuring instructions so as to generate other codes that identify other obscuring instructions that are then selected by the generator.

22. A method of executing a plurality of critical instructions that are obscured by obscuring instructions and stored along with encrypted loading instructions and decryption keys, said method comprising the steps of:

loading into a first memory address a first block of loading instructions,  
executing the loading instructions to allocate a first dynamic memory address,  
loading and executing at the first dynamic memory address at least one critical instruction that is obscured by obscuring instructions,  
retrieving a first decryption key,  
decrypting a second block of loading instructions using the decryption key,  
loading into said first memory address the second block of loading instructions,  
executing the loading instructions to allocate a second dynamic memory address, and  
loading and executing at the second dynamic memory address at least one more critical instruction that is obscured by obscuring instructions.

23. The method of claim 22 comprising the steps of:  
retrieving a second decryption key,  
decrypting a third block of loading instructions using the second decryption key,  
loading into said first memory address the third block of loading instructions,  
executing the loading instructions to allocate a third dynamic memory address, and  
loading and executing at the third dynamic memory address at least one more critical instruction that is obscured by obscuring instructions.